Trabalho Prático de Programação Natural
Cubo Mágico

Gabriel de Biasi[1]

[1] Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627 – Pampulha – Belo Horizonte – MG

biasi@dcc.ufmg.br

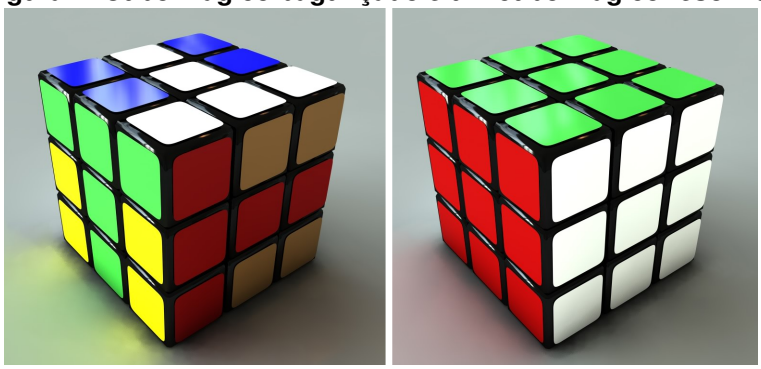
1. Descrição do Problema

O brinquedo "Cubo Mágico" ou *Rubik's Cube* foi criado por Ernő Rubik no ano de 1974 e distribuído comercialmente em 1980. Possui um conjunto de 26 cubos menores, 6 faces com três tipos de cubos distintos: São 4 centros, 12 meios e 8 quinas. Cada tipo cubo possui um número específico de cores, onde os centros possui uma cor, meios possuem duas cores e as quinas possuem três cores.

É possível realizar movimentos em faces do cubo mágico sendo possível rotacionar os cubos desta face. Há 3 movimentos distintos que podem ser feitos em cada face, sendo eles: rotação horária, rotação anti-horária e dupla rotação. Logo, o cubo mágico tem um total de 18 movimentos possíveis.

O objetivo principal deste jogo é fazer com que todas as faces tenham a mesma cor. Na Figura 1 temos na esquerda um cubo em um estado "bagunçado" e outro no estado resolvido.

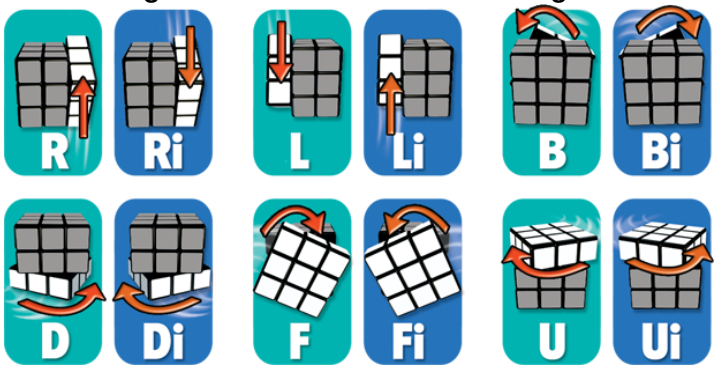
Figura 1. Cubo mágico bagunçado e um cubo mágico resolvido



O mapeamento dos movimentos possíveis no cubo mágico está de acordo com a Figura 2. Os movimentos horários nas faces da esquerda, direita, atrás, baixo, frente e topo serão, respectivamente, L, R, B, D, F e U. Ao acrescentar a letra i, o movimento se torna anti-horário e ao acrescentar o número 2, o movimento se tornará duplo.

Foi provado matematicamente em 2010 que um cubo mágico em qualquer estado pode ser resolvido com 20 movimentos ou menos. Os cálculos foram realizados por um super computador fornecido pela Google. Detalhes sobre este projeto é encontrado no site do projeto: `http://www.cube20.org/`.

Figura 2. Movimentos do cubo mágico



Neste trabalho, é proposto um algoritmo evolutivo que tenha capacidade de resolver uma dada instância de cubo mágico colocando-o no estado resolvido e ao mesmo tempo buscando minimizar a quantidade de movimentos necessários.

2. Metodologia

Para alcançar o objetivo do jogo, foi utilizado como base um método de resolução criado pelo professor *Morwen Thistlethwaite*, onde o espaço de buscas de soluções é categorizado e então o cubo precisa ser levado de uma categoria para a próxima utilizando apenas os movimentos permitidos da categoria atual [Scherphuis 2016].

2.1. Categorias do Cubo

Thistlethwaite criou 5 categorias, que descreve o tão próximo um cubo mágico está da solução. As categorias diminuem drasticamente o espaço de busca de soluções, fazendo com que a busca pela solução simplifique com o avanço entre as categorias.

G0 Todos os estados do cubo mágico possíveis. Naturalmente, todos os cubos mágicos já estão presentes no conjunto G0. Sua ordem é de $|G0| = 4.33 \times 10^{19}$.

G1 Nesta categoria, os meios do cubo estão devidamente **orientados**, ou seja, não são permitidos os movimentos simples $[L, R]$ para colocá-los em sua posição original. Sua ordem é de $|G1| = 2.11 \times 10^{19}$.

G2 Na categoria G2 os meios da camada do meio não podem ser movidos de suas faces e a face de cima e a face de baixo só possuem suas respectivas cores amarelo/branco. Sua ordem é de $|G2| = 1.95 \times 10^{10}$.

G3 Todas as faces opostas do cubo mágico agora possuem suas determinadas cores. As faces frente/atrás terão laranja ou vermelho, cima/baixo terão amarelo ou branco e esquerda/direita terão verde ou azul. Sua ordem é de $|G3| = 6.63 \times 10^{5}$.

G4 Estado do cubo resolvido, $|G4| = 1$.

Os movimentos definidos por Thistlethwaite para cada categoria permitem levar para a próxima sem a quebra de propriedade. Este método de resolução facilita o estilo que um algoritmo evolutivo funciona, fazendo com que toda a população ganhe movimentos diferentes, entretanto, todos ainda pertencem à uma mesma categoria. Os movimentos permitidos são os descritos na Tabela 1.

Tabela 1. Movimentos permitidos em cada categoria

Categoria	Conjunto de Movimentos Permitidos
G0	(F, R, U, B, L, D)
G1	$(F, R, U, B, L2, D2)$
G2	$(F, R, U2, B2, L2, D2)$
G3	$(F2, R2, U2, B2, L2, D2)$
G4	$\emptyset$

3. Descrição Geral da Implementação

[El-Sourani et al. 2010] propuseram em seu artigo a utilização do método de Thistlethwaite porém com algumas modificações nas regras de movimentos e definição das funções de $fitness$.

Neste trabalho, foi decidido manter os movimentos originais do trabalho de Thistlethwaite e as funções de $fitness$ e mutação precisaram ser adaptadas. A estrutura do algoritmo é explicada nas próximas subseções.

3.1. Fluxo de Trabalho do Algoritmo

O algoritmo inicia com a definição de três constantes, $Alpha$ (α), $Theta$ (Θ) e $Lambda$ (λ). Estas constantes representam, respectivamente, a quantidade de gerações máxima do algoritmo, o tamanho da população e a quantidade de indivíduos resolvidos necessária para avançar uma categoria.

A população inicial de tamanho Θ é criada e todos os indivíduos estão vazios. Portanto, todos os indivíduos sofrem sua primeira mutação para iniciar as gerações.

No início da geração, os indivíduos são ordenados por ranqueamento e os λ melhores indivíduos são considerados os candidatos para a reprodução. Se todos os candidatos já pertecerem à próxima categoria, o algoritmo todo avança para a próxima categoria mudando o comportamento da função $fitness$ e de mutação. Todos os cadidatos possuem a mesma probabilidade de se reproduzirem e criar a próxima população, ao fim do processo, todos os novos indivíduos sofrem mutação e iniciam a próxima geração.

O algoritmo encerra quando um indivíduo chega na categoria G4 ou quando o número de gerações definido por α for alcançado.

3.2. Definição de um Indivíduo

Neste algoritmo, um indivíduo é representado por uma simples lista de movimentos no cubo mágico, partindo do estado do cubo que foi passado quando o algoritmo iniciou.

É importante ressaltar que cada indivíduo deste algoritmo evolucionário possui **quatro** valores de $fitness$ diferentes, que é melhor explicado na Subseção 3.6.

3.3. Seleção

A seleção de indivíduos é feita por ranqueamento simples, onde todos os todos indivíduos da população são ordenados pelo seu $fitness$ da categoria atual. Os λ primeiros indivíduos são selecionados e considerados os candidatos para gerar a próxima população.

Após a seleção, todos λ candidatos terão a probabilidade $\frac{1}{\lambda}$ de serem escolhidos e então serem duplicados. Este processo se repete até que a nova população alcance a quantidade de Θ indivíduos.

3.4. Mutação

Ao criar a população inicial, todos os Θ indivíduos são vazios. O único processo que os modifica neste algoritmo é a mutação. De acordo com a categoria atual dos indivíduos da população, eles recebem uma sequência de movimentos diferentes. Ao iniciar uma mutação, é sorteado a quantidade de movimentos que serão concatenados, com uma quantidade mínima e máxima definida.

Na Tabela 2, está classificado a quantidade de movimentos que um indivíduo pode receber de acordo com cada categoria. Essas quantidades foram devidamente calculadas e definidas por Thistlethwaite.

Tabela 2. Exemplos de Indivíduos

Categoria	Quantidade de Movimentos
$G0 \rightarrow G1$	$0 \leq l \leq 7$
$G1 \rightarrow G2$	$1 \leq l \leq 13$
$G2 \rightarrow G3$	$2 \leq l \leq 15$
$G3 \rightarrow G4$	$4 \leq l \leq 17$

Ao criar a nova sequência de movimentos, é provável que alguns movimentos possam ser removidos por serem complementares ou não produzir efeito final no cubo mágico. Portanto, todas as vezes que a mutação produz uma sequência de movimentos, ela é verificada pela função $clean$ em busca destes tipos de movimentos.

3.5. Função $clean$

Esta função é chamada toda vez que uma sequência de mutação é criada para um indivíduo. A fim de reduzir o número de movimentos necessários para resolver o cubo mágico, movimentos em sequência que não geram efeitos no cubo são removidos e movimentos que podem ser simplificados são alterados, sem perda de contexto final do cubo.

A seguir, temos três exemplos de simplificação que podem ser feitos em uma sequência de movimentos. Na Tabela 3 apenas o movimento F é apresentado, entretanto estas regras podem ser utilizadas para quaisquer movimentos do cubo mágico e em qualquer ordem.

Tabela 3. Exemplos de Reduções de Movimentos

Inicial	Motivo	Final
$[F, Fi]$	Não produz efeito	$[\emptyset]$
$[F, F]$	Torna-se um giro duplo	$[F2]$
$[F, F2]$	Torna-se um giro invertido	$[Fi]$

3.6. Função $fitness$

Para adequar-se ao estilo de categorias, foi necessário criar quatro funções de $fitness$ diferentes, sendo uma para cada mudança de categoria. Nas próximas subseções, é apresentado cada uma delas.

Todas as condições que são apresentadas nas próximas seções são ruins, ou seja, na implementação o valor do $fitness$ recebe acrécimo de valor por condição satisfeita e o objetivo dos indivíduos é minimizar este valor.

Para fins de padronização, o cubo mágico que estamos trabalhando tem a frente com a cor laranja, atrás com a cor vermelha, topo com a cor amarela, baixo com a cor branca, esquerda com a cor verde e a direita com a cor azul.

3.6.1. Mudança de $G0 \rightarrow G1$

Neste momento, precisamos verificar se os movimentos dos indivíduos deixam o cubo no estado G1, onde os meios estão orientados. Logo, é feita as seguintes verifições nas faces **frente, atrás, esquerda e direita**:

- Cor amarela/branca nos meios da face;
- Cor azul/verde nos meios da face porém junto com laranja/vermelha nos lados.

Se algum destes casos forem verdadeiros, o cubo ainda não pertence à categoria G1. Esta fase do algoritmo tende à ser a mais rápida, pois no pior caso são necessários apenas sete movimentos para orientar os meios.

3.6.2. Mudança de $G1 \rightarrow G2$

Como os cubos G1 já estão com os meios orientados, nesta fase os meios que pertencem à camada do meio precisam ser colocados lá, porém não é necessário colocá-lo em seu local exato, apenas na camada.

Para colocar os cubos meio na camada do meio é feita a seguinte verificação nas faces **frente, atrás, esquerda e direita**:

- Cubos meio da camada do meio fora da camada;

Agora, nesta fase os cantos também precisam ser orientados. Para orientar os cantos, é feita a seguinte verificação nas faces **topo e baixo**:

- Cores azul/verde nos cantos da face;
- Cores vermelho/laranja nos cantos da face.

Se algum destes casos forem verdadeiros, o cubo ainda não pertence à categoria G2. Esta fase do algoritmo é um pouco mais lenta, pois trabalha com dois processos distintos. Na implementação real, este processo foi dividido em 2 sub-fases para melhor convergência.

3.6.3. Mudança de $G2 \rightarrow G3$

Os cubos na categoria G2 possuem suas faces topo e baixo com suas cores amarelo/branco. Agora, é necessário fazer o mesmo para o resto das faces. Portanto, para alcançar a categoria G3, é feita a seguinte verificação nas faces **frente, atrás, esquerda e direita**:

- Cores vermelho/laranja na faces esquerda/direita;
- Cores azul/verde nas faces frente/atrás.

Se algum destes casos forem verdadeiros, o cubo ainda não pertence à categoria G3. Esta fase do algoritmo também é lenta, pois são necessários muitos movimentos para as cores irem as suas respectivas faces.

3.6.4. Mudança de $G3 \rightarrow G4$

Neste ponto, todos os cubos estão na categoria G3 e possuem apenas cores corretas e cores opostas em suas faces. É necessário agora apenas os movimentos duplos para concluir o cubo mágico. Para avançar para a categoria G4, é feita a seguinte verificação em **todas** as faces:

- Cores que não combinam com a cor do centro da face.

Se este caso for verdadeiro, o cubo ainda não pertence à categoria G4. Esta fase do algoritmo vários movimentos duplos são necessários para resolvê-lo. Também é considera uma fase lenta. Entretanto, os indivíduos que passarem por esta fase concluem o objetivo do jogo.

4. Execução dos Experimentos

lala

5. Conclusão

Neste trabalho.

Referências

[El-Sourani et al. 2010] El-Sourani, N., Hauke, S., and Borschbach, M. (2010). An evolutionary approach for solving the rubik's cube incorporating exact methods.

[Scherphuis 2016] Scherphuis, J. (2016). Thistlethwaite's 52-move algorithm.